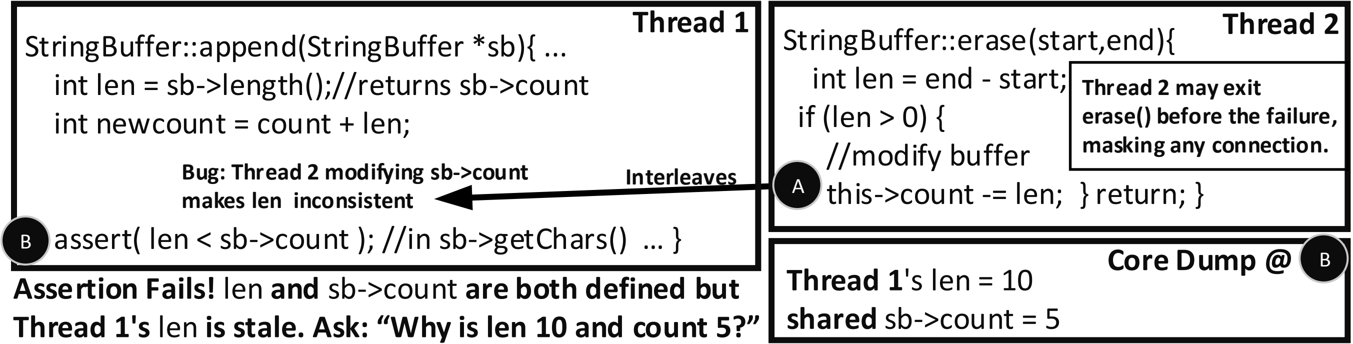

Thread 1

```
StringBuffer::append(StringBuffer *sb){ ...
   int len = sb->length();//returns sb->count
   int newcount = count + len;

B  assert( len < sb->count ); //in sb->getChars()  ... }
```

Bug: Thread 2 modifying sb->count makes len inconsistent

Interleaves

Thread 2

```
StringBuffer::erase(start,end){
   int len = end - start;
  if (len > 0) {
    //modify buffer
A   this->count -= len;  } return; }
```

Thread 2 may exit erase() before the failure, masking any connection.

Core Dump @ B

Thread 1's len = 10

shared sb->count = 5

Assertion Fails! len **and** sb->count **are both defined but Thread 1's** len **is stale. Ask: "Why is len 10 and count 5?"**